

AD-A142 812

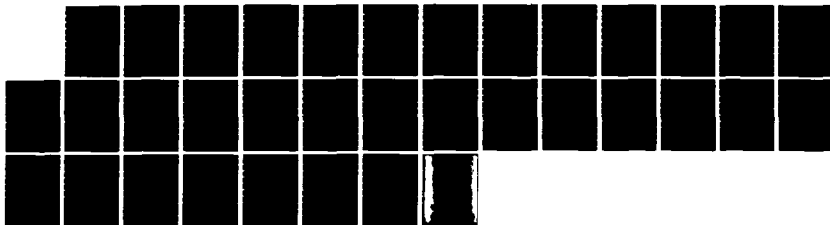
SPEEDING UP THE HELP HYDRODYNAMIC CODE(U) MATERIALS
RESEARCH LABS ASCOT VALE (AUSTRALIA) D L SMITH MAY 84
MRL-R-925

1/1

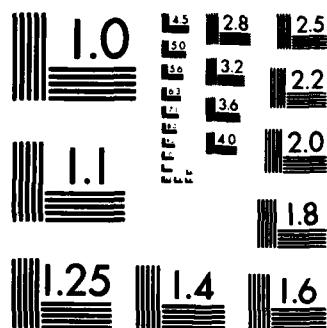
UNCLASSIFIED

F/G 9/2

NL







MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12



DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
MATERIALS RESEARCH LABORATORIES
MELBOURNE, VICTORIA

REPORT

MRL-R-925

SPEEDING UP THE HELP HYDRODYNAMIC CODE

David L. Smith

THE UNITED STATES NATIONAL
TECHNICAL INFORMATION SERVICE
IS AUTHORIZED TO
REPRODUCE AND SELL THIS REPORT

Approved for Public Release



DTIC
ELECTE
S **D**
JUL 1 0 1984
E

DTIC FILE COPY

AD-A142 812

84 07 09 001

MAY, 1984

DEPARTMENT OF DEFENCE
MATERIALS RESEARCH LABORATORIES

REPORT
MRL-R-925

SPEEDING UP THE HELP HYDRODYNAMIC CODE

David L. Smith

ABSTRACT

The Eulerian hydrodynamic code HELP has been employed at MRL for some years to model problems in warhead research. One disadvantage of HELP has been the long computer run-times associated with its use. This report describes a modification to the code which makes it run up to three times faster.

Approved for Public Release

POSTAL ADDRESS: Director, Materials Research Laboratories
P.O. Box 50, Ascot Vale, Victoria 3032, Australia

DOCUMENT CONTROL DATA SHEET

REPORT NO.
MRL-R-925AR NO.
AR-003-907REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

TITLE

SPEEDING UP THE HELP HYDRODYNAMIC CODE

AUTHOR(S)

DAVID L. SMITH

CORPORATE AUTHOR

Materials Research Laboratories
P.O. Box 50,
Ascot Vale, Victoria 3032REPORT DATE
MAY, 1984TASK NO.
DST 82/192SPONSOR
DSTO

CLASSIFICATION/LIMITATION REVIEW DATE

CLASSIFICATION/RELEASE AUTHORITY
Superintendent, MRL
Physical Chemistry Division

SECONDARY DISTRIBUTION

Approved for Public Release

ANNOUNCEMENT

Announcement of this report is unlimited

KEYWORDS

Shaped charges
Computer programsHELP Hydrodynamic code
Collapsing liner warheads
Explosive-metal interactions

COSATI GROUPS

0902

1904

ABSTRACT

The Eulerian hydrodynamic code HELP has been employed at MRL for some years to model problems in warhead research. One disadvantage of HELP has been the long computer run-times associated with its use. This report describes a modification to the code which makes it run up to three times faster.

C O N T E N T S

	<u>Page No.</u>
1. INTRODUCTION	1
2. ANALYSIS OF HELP PERFORMANCE	1
2.1 Run-time Profiles by Subroutine	1
2.2 Analysis of Subroutine MOVTCR	2
3. MODIFICATIONS TO MOVTCR	3
4. RESULTS	4
5. REFERENCES	4
APPENDIX 1 - Job to Run Hemispherical Shaped Charge Problem	
APPENDIX 2 - Job to Run Shaped Charge Problem	
APPENDIX 3 - Listing of Subroutine MOVTCR	
APPENDIX 4 - An Update Run to Add Subroutine TCRCHK to the HELP Code	

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
_____ and/or	

A-1	

SPEEDING UP THE HELP HYDRODYNAMIC CODE

1. INTRODUCTION

The two-dimensional Eulerian hydrodynamic computer code HELP (Hydrodynamic Elastic Plastic) has been employed at the Materials Research Laboratories (MRL) for some years to model explosive-metal interactions in warhead research. HELP was developed in the US during the 1970's, and is well described in Reference [1]. The version of HELP in use at MRL [2] was obtained from the US Army Ballistic Research Laboratory through the auspices of TTCP.

HELP suffers the common disadvantage of Eulerian codes in that it demands long computer run-times to solve useful problems. At MRL, HELP has largely been run on a CDC-7600 computer on the CSIRONET network. For warhead problems such as shaped charge collapse, 3 to 4 hours CPU time on the 7600 would be required for a typical run. This is not only expensive but also inconvenient, as run-times of this length exceed user limits on CSIRONET, and can only be achieved by stopping and restarting at a later date.

With these factors in mind, it was decided to investigate the possibility of improving the efficiency of HELP, without changing its basic *modus operandi*. This report describes the analysis of the code's run-time performance, and presents a modification to HELP which increases its speed up to threefold.

2. ANALYSIS OF HELP PERFORMANCE

2.1 *Run-time Profiles by Subroutine*

The first approach to analysing the performance of HELP was to establish which of the subroutines were absorbing most of the computer time. This was achieved using the P-register sampling utilities (PREGON, PRLIST)

available on the 7600 under the SCOPE operating system [3]. These utilities record the programme address under execution every 3.6 milliseconds, when the hardware clock interrupts the executing programme for system housekeeping. At the end of execution, these data are processed to build a statistical profile of programme addresses vs frequency. A loader map can then be used to associate these addresses with programme segments.

Two fairly typical problems previously run with HELP at MRL were chosen and re-run with P-register analysis. The first of these was a 1.5 inch diameter hemispherical shaped charge, given in Reference [2] as a sample problem for the BRL version of HELP. The complete input deck for this run is given in Appendix 1. (The input has been modified slightly from that in Reference [2], partly due to memory constraints, and partly to include the strength phase of HELP to assess its time consumption). Schematic illustrations for this problem at the beginning and end of the HELP run may be found in Figures 1 and 2 respectively. The second study chosen was a 38 mm shaped charge of the type used frequently in experimental studies at MRL [4]. The input deck for this run is given in Appendix 2, and illustrations are at Figures 3 and 4.

The profile analyses for these two jobs are presented in terms of the percentage of total CPU time taken by each subroutine. The hemisphere job, run 1.023, used 5720 seconds of 7600 time, and its analysis is shown in Table 1. The shaped charge job, run 50.0009, was stopped after 500 cycles and used 6829 seconds CPU time; its analysis is shown in Table 2.

The profiles reveal that in both runs the tracer moving subroutine MOVTCR accounted for the bulk of the run-time. This routine used 61.7% of the total time in the first run, and 68.8% in the second. Clearly a significant improvement to the overall performance of HELP could only be achieved by improving MOVTCR, and hence this subroutine was analysed in detail.

2.2 Analysis of Subroutine MOVTCR

Subroutine MOVTCR is called on every sub-cycle of HELP, to move the massless tracer particles which define material boundaries. MOVTCR also moves cell-centred passive tracers if these have been selected. Both types of tracer are moved in a pseudo - continuous local velocity field: the method is described in detail in Section 4.1.2 of Reference [1].

In order to establish exactly which segments of MOVTCR were consuming the computer time, a further P-register analysis run was performed. This run concentrated on the load addresses occupied by MOVTCR. A cross-referenced listing then enabled identification of the offending code. It was found that over 95% of the subroutine's time was spent in one small inner loop. Appendix 3 contains a listing of MOVTCR: the loop referred to is DO 1200, between MOVTRC.48 and MOVTCR.52.

Understanding the function of this loop requires a little knowledge of the workings of HELP. Each material package is defined spatially by an

ordered set of tracer particles which form the package boundaries. These tracers are also used by the code to "sense" whether material is entering or leaving a cell, and to define fractional cell areas across which such transport may occur. This is quite the most complicated logic in HELP. The logic requires that the "void package", present in nearly all HELP problems, have its own set of tracer particles. These tracers must be coincident with tracers from other material packages but are kept as an independent ordered set. When tracer particles for (non-void) materials are moved on each cycle or sub-cycle, the corresponding void tracer is found and moved identically. The function of the DO 1200 loop in MOVTCR is simply to find whether the tracer being considered has a corresponding void tracer which needs to be moved. Since most typical HELP runs involve in the order of 1000 tracers for each package, it is not surprising that this loop, being executed each sub-cycle in the order of 1,000,000 times, proved very time-consuming.

3. MODIFICATIONS TO MOVTCR

Since coincident tracers are moved identically, it is not necessary to check for these every cycle or sub-cycle, because they will remain associated. The only way this association could alter is when new tracers are added or existing ones removed. This can only happen when subroutine ADDTCR is called to interpolate new tracers because existing ones have become sparse, or when tracers are adjusted manually on restarts or automatically by VDCLOS, the void closing routine.

The solution was to associate each void package tracer with a corresponding material tracer and store this information at the time of problem set-up. Further checks then need only be made if tracers have been added or deleted.

To do this, a new subroutine TCRCHK was added to HELP. This subroutine checks each tracer of the void package to find a coincident material package tracer. A "pointer" array is filled so that each void tracer has a floating point number associated with it. The fractional part of this number contains the material package number, and the integer part the tracer number for the matching tracer. Subroutine MOVTCR now moves all non-void tracers first, and then moves void tracers according to the information encoded in this array. TCRCHK is called on problem initialisation and then only called again if ADDTCR adds new tracers, if VDCLOS removes tracers, or on problem restart. An UPDATE [5] correction run to incorporate these modifications, including the new routine TCRCHK, is given in Appendix 4.

Whilst it is not claimed that this solution is necessarily the most elegant, its attraction is that it makes no changes to the existing tracer particle logic in HELP and hence cannot have any unforeseen consequences. It should also substantially reduce run-times for all HELP simulations which include a void package.

4. RESULTS

The two test problems chosen in Section 2.1 were re-run with the modified version of HELP. Resultant outputs were found to be identical. For the hemisphere calculation, run-time was reduced from 5720 seconds to 2440 seconds (43%), and for the shaped charge calculation from 6829 seconds to 2386 seconds (35%). Tables 3 and 4 show subroutine profile analyses for these runs. For the hemisphere run subroutines MOVTCR and the new routine TCRCHK together accounted for 6.9% of the run-time, compared to the previous figures of 61.7%. For the shaped charge run the corresponding figures are 7.6% and 68.8%.

It seems reasonable to assume that for most applications of HELP, this modification will increase the code's speed by a factor of between two and three.

5. REFERENCES

1. HELP - A Multimaterial Eulerian Program in Two Space Dimensions and Time, L.J. Hageman et al, Systems, Science and Software. Final Report AFATL-TR-76-45, April 1976.
2. The BRL 7600 Version of the HELP Code. J. Lacetera et al, US Army Ballistic Research Laboratory, Technical Report ARBRL-TR-02209, January 1980.
3. SCOPE 2.1 Reference Manual, Revision 2, Control Data Corporation, Manual 60342600.
4. Fabrication and Performance Assessment of Shaped Charges. M.C. Chick and G.P. Briggs, Defence Standards Laboratories Report No. 546, June 1974.
5. UPDATE Reference Manual, Control Data Corporation, Manual 60449900 (UPDATE is a utility for maintaining large files).

T A B L E 1

PROFILE ANALYSIS FOR RUN 1.023

RUN-TIME = 5720 seconds

(PERCENTAGES TO NEAREST 0.1%)

SUBROUTINE	% TIME	SUBROUTINE	% TIME
MAIN	0.0	NEWMIX	0.0
ADDTCR*	0.0	NEWRHO	0.0
CALFRC	0.0	PTSAV	0.1
CDT	2.8	REZONE	0.0
CMPSN	0.0	RNDOFF	0.0
COMPRS	0.0	SETUP	0.0
DETME	0.0	SETUPA	0.2
DMADJ	0.8	SPHASE	2.6
DMCALC	1.7	STRNG	0.1
EDIT	0.0	THETAS	0.1
ENCHK	3.3	TPHASE	11.1
ENDMV	0.0	TSETUP	0.0
EOUT	0.1	UVCALC	0.0
EQST	1.3	UVMOD	0.2
ERROR	0.0	VDCLOS	0.0
FILGRD	0.0	XY,Y,-PTOC	0.1
FLGSET	1.9	ADDENG	0.0
FRACS	1.9	UC	0.0
HPHASE	5.8	VC	0.0
INFACE	0.0	VOLFND	0.0
INPUT	0.0	WN	0.0
MAP	0.1	X,Y,-CTOP	0.0
MOVTCR	61.7	SYSTEM FUNCTIONS	3.4
NEWFLG	0.0		

* No tracers added in this run.

T A B L E 2

PROFILE ANALYSIS FOR RUN 50.0009

RUN-TIME = 6829 SECONDS

(PERCENTAGES TO NEAREST 0.1%)

SUBROUTINE	% TIME	SUBROUTINE	% TIME
MAIN	0.0	NEWMIX	0.0
ADDTCR	13.3	NEWRHO	0.0
CALFRC	0.0	PTSAV	0.2
CDT	1.3	REZONE	0.0
CMPSN	0.0	RNDOFF	0.0
COMPRS	0.0	SETUP	0.0
DETME	0.0	SETUPA	0.2
DMADJ	0.5	SPHASE*	0.0
DMCALC	1.3	STRNG*	0.0
EDIT	0.0	THETAS	0.3
ENCHCK	1.3	TPHASE	4.5
ENDMV	0.0	TSETUP	0.0
EOUT	0.0	UVCALC	0.1
EQST	0.6	UVMOD	0.0
ERROR	0.0	VDCLOS	0.0
FILGRD	0.0	X, Y, -PTOC	1.4
FLGSET	0.8	ADDENG	0.0
FRACS	1.8	UC	0.0
HPHASE	1.8	VC	0.0
INFACE	0.0	VOLFND	0.0
INPUT	0.0	WN	0.0
MAP	0.0	X, Y, -CTOP	0.0
MOVTCR	68.8	SYSTEM FUNCTIONS	1.3
NEWFLG	0.0		

* No strength phase in this run.

T A B L E 3

PROFILE ANALYSIS FOR RUN 1.024

RUN-TIME = 2440 SECONDS

(PERCENTAGES TO NEAREST 0.1%)

SUBROUTINE	% TIME	SUBROUTINE	% TIME
MAIN	0.0	NEWMIX	0.0
ADDTCR*	0.0	NEWRHO	0.0
CALFRC	0.0	PTSAV	0.1
CDT	6.7	REZONE	0.0
CMPRSN	0.0	RNDOFF	0.0
COMPRS	0.0	SETUP	0.0
DETIME	0.0	SETUPA	0.4
DMADJ	2.0	SPHASE	6.3
DMCALC	4.5	STRNG	0.1
EDIT	0.1	THETAS	0.1
ENCHCK	9.1	TPHASE	26.7
ENDMV	0.0	TSETUP	0.0
EOUT	0.0	UVCALC	0.1
EQST	3.2	UVMOD	0.5
ERROR	0.0	VDCLOS	0.0
FILGRD	0.0	X, Y, -PTOC	1.7
FLGSET	4.4	ADDENG	0.0
FRACS	4.6	UC	0.0
HPHASE	13.8	VC	0.0
INFACE	0.0	VOLFND	0.0
INPUT	0.0	WN	0.0
MAP	0.2	X, Y, -CTOP	0.0
MOVTCR	6.8	TCRCHK	0.1
NEWFLG	0.1	SYSTEM FUNCTIONS	8.2

* No tracers added in this run

T A B L E 4

PROFILE ANALYSIS FOR RUN 50.0010

RUN-TIME = 2386 SECONDS

(PERCENTAGES TO NEAREST 0.1%)

SUBROUTINE	% TIME	SUBROUTINE	% TIME
MAIN	0.0	NEWMIX	0.0
ADDTCR	39.7	NEWRHO	0.0
CALFRC	0.0	PTSAV	0.8
CDT	3.6	REZONE	0.0
CMPSRN	0.0	RNDOFF	0.0
COMPRS	0.0	SETUP	0.0
DETIME	0.0	SETUPA	0.6
DMADJ	1.3	SPHASE*	0.0
DMCALC	4.0	STRNG*	0.0
EDIT	0.1	THETAS	0.7
ENCHCK	4.4	TPHASE	13.2
ENDMV	0.0	TSETUP	0.0
EOUT	0.0	UVCALC	0.2
EQST	1.6	UVMOD	0.1
ERROR	0.0	VDCLOS	0.0
FILGRD	0.0	ADDENG	0.0
FRACS	5.4	UC	0.0
HPHASE	5.5	VC	0.0
INFACE	0.0	VOLFND	0.0
INPUT	0.0	WN	0.0
MAP	0.1	X, Y, -CTOP	0.0
MOVTCR	5.8	TCRCHK	1.8
NEWFLG	0.1	SYSTEM FUNCTIONS	4.0

* No strength phase in this run

TIME = .0 CYCLE = 0

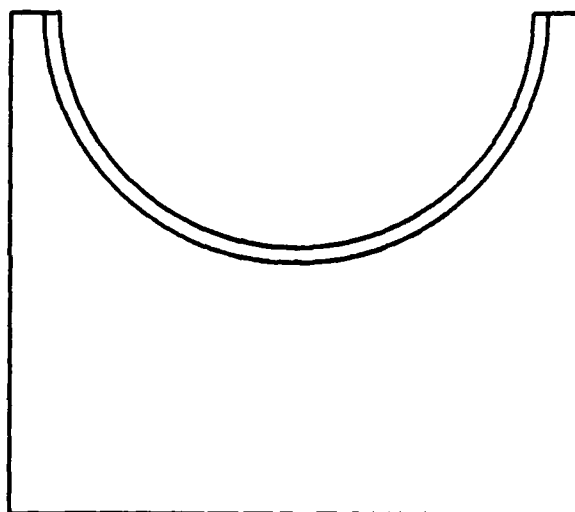


FIGURE 1

Hemispherical Shaped Charge

TIME = 16.0 CYCLE = 644

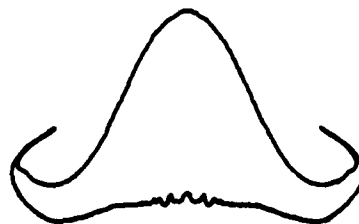


FIGURE 2

Hemispherical Shaped Charge

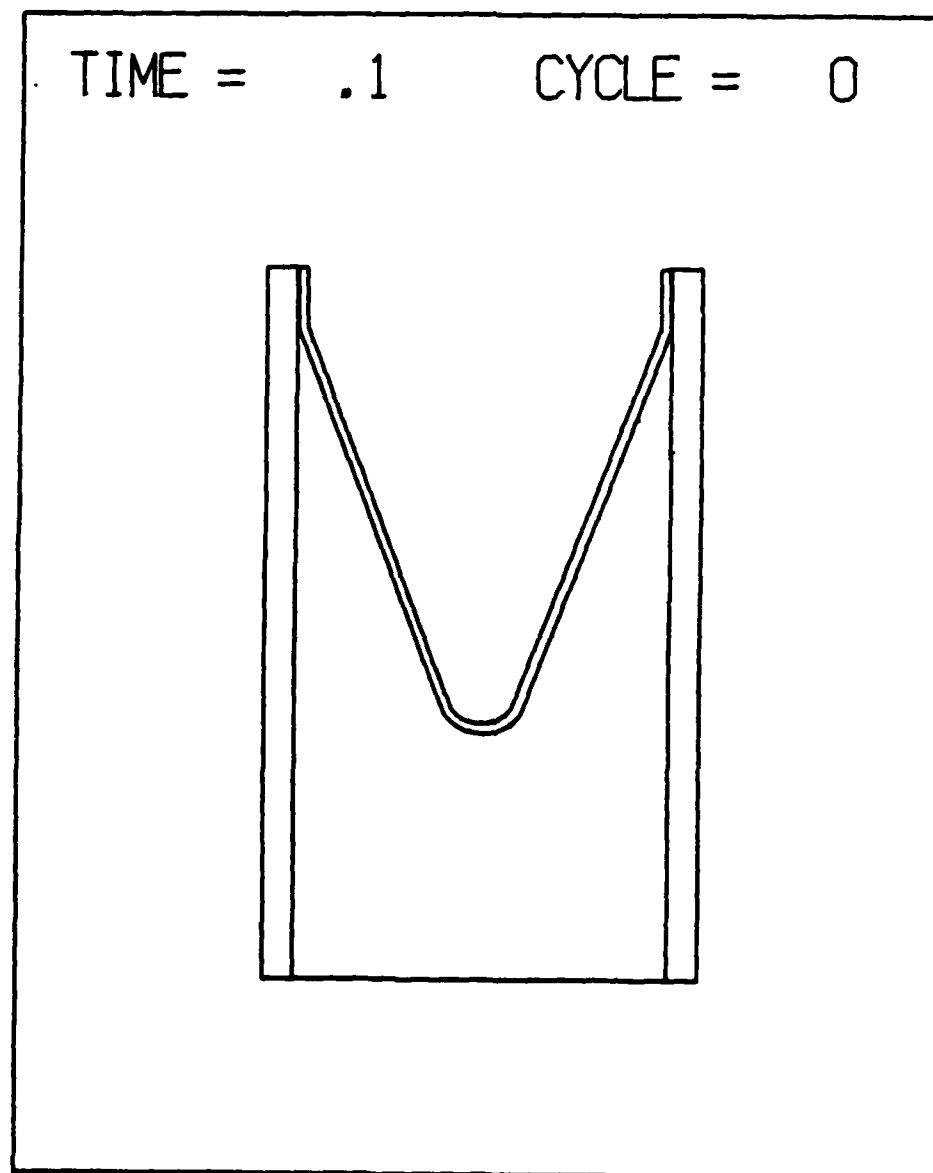


FIGURE 3
MRL Shaped Charge

TIME = 9.2

CYCLE = 500

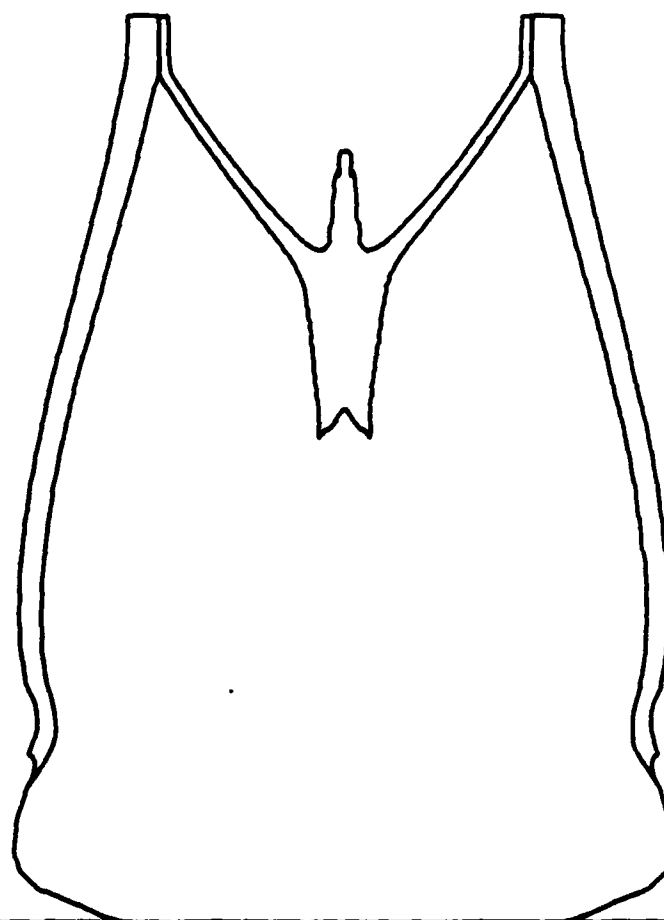


FIGURE 4

MRL Shaped Charge

APPENDIX 1

JOB TO RUN HEMISPHERICAL SHAPED CHARGE PROBLEM

```

SCRUN(P10,T20000,MS140000,ML600)
COMMENT. JOB TO RUN PROBLEM 1.023
COMMENT. SAMPLE PROBLEM OF HEMISPHERE
LIMIT(0)
GETSET(DFC4405)
REQUEST(TAPE1,*PF,SN=DFC4405)
REQUEST(TAPE7,*PF,SN=DFC4405)
REQUEST(TAPE20,*PF,SN=DFC4405)
ATTACH,LGO,$LGO FOR MRLHLP20$,SN=DFC4405,ID=DFCDLS.
MAP(PART)
PREGON(FWA=0,LWA=140000)
LGO(PL=200000)
EXIT(U)
PRLIST(,TAPE20,)
CATALOG(TAPE20,$PRLIST JOB 1.023$,SN=DFC4405,ID=DFCDLS,RP=998)
REWIND(TAPE20)
COPY(TAPE20,OUTPUT)
EXIT(U)
REWIND(TAPE7)
ATTACH(LGO2,$LGO FOR FICHE PLOTS$,SN=DFC4405,ID=DFCDLS)
MAP(OFF)
LGO2.
EXIT(U)
REWIND(TAPE7)
CATALOG(TAPE7,$TAPE7 JOB 1.023$,SN=DFC4405,ID=DFCDLS,RP=998)
EXIT(U)
DISPOSE(FICHE,*PR=CFC,ST=COMNM)
DISPOSE(FICHE2,*FL=CFC,ST=COMNM)
REQUEST(FRED,*PF,SN=DFC4405)
FUSE.
TITLE(FICHE)
TITLE(FICHE2)
TRANZP(TAPE1,FL,FID=FICHE2)
CATALOG(TAPE1,$ZP FILE JOB 1.023$,SN=DFC4405,ID=DFCDLS,RP=999)
EXIT(U)
REWIND(OUTPUT)
COPY(OUTPUT,FICHE)
EXIT(U)
REWIND(OUTPUT)
COPY(OUTPUT,FRED)
CATALOG(FRED,$OUTPUT JOB 1.023$,SN=DFC4405,ID=DFCDLS,RP=20)
EXIT(U)
DISPOSE(OUTPUT,SC).
*EOS
SAMPLE PROBLEM 2 - COPPER HEMISPHERE LOADED WITH COMP B HE UNCONFINED
$OPTNS NPRTOP=2, TIMMAX=8000, IFLGST=1, IPMADJ=1, ITPHSE=1$
$START PK(1)=1.02, PK(2)=0, PK(3)=-0$
$RUN TSTOP=1.5E-04, PRDEL=2.0E-06, KUNITW=7, KUNITR=7, NOSLIP=0,
LVISC=1, NMAT=2, NMXCLS=600, NTRACR=5, NTPMX=900, REZ=0.,
CYCPH3=1, NLINER=1, NSLD=300, MAPS=1, IMAX=60, JMAX=132,
DMIN=100, IPR=100, NRELP=100, NDUMP7=1, IPCYCL=0, I1=3,
I2=53, CVIS=-1, ICM=0$
60 .05
999
132 .06666667
999

```

0						
2	8.90	0.0	0.0	0.0		
21	1.717	0.0	0.0	0.0		
2.35E+09	6.95E+10	5.50E+10	5.30E+09	4.55E+11	.9785	
0.0	0.0	0.0	0.0	0.0	1.0	
4	1	300				
3.14159	1.57080					
0.0	6.35	1.905				
1	1	50				
1.905	6.35	1.7907	6.35			
-4	1	300				
1.57080	3.14159					
0.0	6.35	1.7907				
1	2	100				
0.0	2.54	2.159	2.54			
2	2	200				
2.159	2.54	2.159	6.35			
1	2	50				
2.159	6.35	1.905	6.35			
-4	2	300				
1.57080	3.14159					
0.0	6.35	1.905				
4	3	300				
3.14159	1.57080					
0.0	6.35	1.7907				
1	3	50				
1.7907	6.35	1.905	6.35			
1	3	50				
1.905	6.35	2.159	6.35			
2	3	200				
2.159	6.35	2.159	2.54			
-1	3	100				
2.159	2.54	0.0	2.54			
100						
1	0	1	0	300	0	
0	2	0	351	0	650	
	1		2		0.0	2.54
	2		2		1.82	5.62
	0					
0.0		2.54	2.159	6.35		
0.0		5.59	2.159	6.35		

*EOS

LIST.

TOTAL(0.140000)

*EOS

TITLE1=\$OUTPUT JOB 1.023\$

TITLE2=\$DAVID L. SMITH\$

TITLE3=\$WRITTEN "DATE"\$

INDEX=1/1/25

*EOS

TITLE1=\$PLOTS FOR JOB 1.023\$

TITLE2=\$DAVID L. SMITH\$

TITLE3=\$WRITTEN "DATE"\$

PLOT

*EOP

APPENDIX 2

JOB TO RUN SHAPED CHARGE PROBLEM

```

SCRUN(P10,T20000,MS140000,ML600)
COMMENT. JOB TO RUN PROBLEM 50.0009
COMMENT. PLANE WAVE DETONATION, MRL 38 MM CHARGE
LIMIT(0)
GETSET(DFC4405)
REQUEST(TAPE1,*PF,SN=DFC4405)
REQUEST(TAPE7,*PF,SN=DFC4405)
REQUEST(TAPE20,'PF',SN=DFC4405)
ATTACH,LGO,$LGO FOR MRLHLP20$,SN=DFC4405,ID=DFCDLS.
MAP(PART)
PREGON(FWA=0,LWA=140000)
LGO(PL=200000)
EXIT(U)
PRLIST(.TAPE20,)
CATALOG(TAPE20,$PRLIST JOB 50.0009$,SN=DFC4405,ID=DFCDLS,RP=998)
REWIND(TAPE20)
COPY(TAPE20,OUTPUT)
EXIT(U)
REWIND(TAPE7)
ATTACH(LGO2,$LGO FOR FICHE PLOTS$,SN=DFC4405,ID=DFCDLS)
MAP(OFF)
LGO2.
EXIT(U)
REWIND(TAPE7)
CATALOG(TAPE7,$TAPE7 JOB 50.0009$,SN=DFC4405,ID=DFCDLS,RP=998)
EXIT(U)
DISPOSE(FICHE,*PR=CFC,ST=COMNM)
DISPOSE(FICHE2,*FL=CFC,ST=COMNM)
REQUEST(FRED,*PF,SN=DFC4405)
FUSE.
TITLE(FICHE)
TITLE(FICHE2)
TRANZP(TAPE1,FL,FID=FICHE2)
CATALOG(TAPE1,$ZP FILE JOB 50.0009$,SN=DFC4405,ID=DFCDLS,RP=999)
EXIT(U)
REWIND(OUTPUT)
COPY(OUTPUT,FICHE)
EXIT(U)
REWIND(OUTPUT)
COPY(OUTPUT,FRED)
CATALOG(FRED,$OUTPUT JOB 50.0009$,SN=DFC4405,ID=DFCDLS,RP=20)
EXIT(U)
DISPOSE(OUTPUT,SC)
*EOS
SHAPED CHARGE RUN ... MRL STANDARD 38 MM CHARGE
$OPTNS IFLGST=1, IPMAJ=1, ITPHSE=1, NPRTOP=2, TIMMAX=7200$
$START PK(1)=50.0009, PK(2)=0, PK(3)=0$
$RUN BBAR=.5, CRATIO=1.0E04, CVIS=-1, CYCMX=2, CYCPH3=-1,
DMIN=1.0E-03, DTMIN=1.0E-11, EMIN=1.0E07, FINAL=0.4,
IEXTX=0, ICM=0, IMAX=60, IPR=35, I1=3, I2=13, JMAX=132,
KUNITR=7, KUNITW=7, LVISC=1, MAPS=2, MINX=0, MAXX=40,
MINY=27, MAXY=106, NADD=10, NDUMP7=1, NRELP=20,
NMXCLS=600, NLINER=1, NMAT=3, NSLD=300, NTCC=0,
NTPMX=1800, NTRACR=3, PMIN=5.0E06, PRCNT=1.0E-03,
PRDEL=1.0E-06, ROEPS=1.0E-05, SIEMIN=1.0E05, STAB=1.0E-03,
ICSTOP=500$

```

40	1	1	1	0.05	.0550	.0605	.0666
1	1	1	1	.0732	.0805	.0886	.0974
1	1	1	1	.1072	.1179	.1297	.1427
1	1	1	6	.1569	.1726	.1899	.2000
999							
12	1	1	1	.2000	.1899	.1726	.1569
1	1	1	1	.1427	.1297	.1179	.1072
1	1	1	1	.0974	.0886	.0805	.0732
1	1	1	80	.0666	.0605	.0550	.0500
1	1	1	1	.0550	.0605	.0666	.0732
1	1	1	1	.0805	.0886	.0974	.1072
1	1	1	1	.1179	.1297	.1427	.1569
1	1	12		.1726	.1899	.2000	
999							
0							
2	8.9			0.0	0.0	0.0	
21	1.717			0.0	0.0	0.0	
4	2.79			0.0	0.0	0.0	
2.35E+09	6.95E+10	5.50E+10	5.30E+09	4.55E+11		0.9875	
0.0	0.0	0.0	0.0	0.0	0.0	1.0	
3.00E+09	0.0	0.0	7.00E+09	2.74E+11		0.9850	
4	1	50					
3.1416	2.1864						
0.0	4.9972	0.4826					
3	1	300					
0.3940	4.7185	1.905	8.6548				
2	1	30					
1.905	8.6548	1.905	9.2898				
1	1	10					
1.905	9.2898	1.8034	9.2898				
2	1	30					
1.8034	9.2898	1.8034	8.6736				
3	1	300					
1.8034	8.6736	0.3040	4.7675				
-4	1	50					
2.2179	3.1416						
0.0	4.9972	0.3810					
1	2	150					
0.0	2.0	1.905	2.0				
2	2	150					
1.905	2.0	1.905	8.6548				
3	2	300					
1.905	8.6548	0.3940	4.7185				
-4	2	50					
2.1864	3.1416						
0.0	4.9972	0.4826					
1	3	10					
1.905	2.0	2.2225	2.0				
2	3	180					
2.2225	2.0	2.2225	9.2898				
1	3	10					
2.2225	9.2898	1.905	9.2898				
2	3	30					
1.905	9.2898	1.905	8.6548				
-2	3	150					
1.905	8.6548	1.905	2.0				

4	4	50		
3.1416	2.2179			
0.0	4.9972	0.3810		
3	4	300		
0.3040	4.7675	1.8034	8.6736	
2	4	30		
1.8034	8.6736	1.8034	9.2898	
1	4	10		
1.8034	9.2898	1.905	9.2898	
1	4	10		
1.905	9.2898	2.2225	9.2898	
2	4	180		
2.2225	9.2898	2.2225	2.0	
1	4	10		
2.2225	2.0	1.905	2.0	
-1	4	150		
1.905	2.0	0.0	2.0	
100				
1	0	1	0	350
0	2	0	151	0
3	0	231	0	380
	99		2	0.0
0.0	2.0	1.91	2.0	8.66
				0.0

*EOS

LIST.

TOTAL(0.140000)

*EOS

TITLE1=\$OUTPUT JOB 50.0009\$

TITLE2=\$DAVID L. SMITH\$

TITLE3=\$WRITTEN "DATE"\$

INDEX=1/1/25

*EOS

TITLE1=\$PLOTS FOR JOB 50.0009\$

TITLE2=\$DAVID L. SMITH\$

TITLE3=\$WRITTEN "DATE"\$

PLOT

*EOP

LISTING OF SUBROUTINE MOVTCR

```

SUBROUTINE MOVTCR
COMMON/INF/ L,J,ICY
COMMON/MOV/LS,UEFF,VEFF,KI(4),UTK(4),VTK(4),WF(4),MTK(4)

C***** FIT2 COMMON BLOCK FOR MRLHELP *****
COMMON Z(1)
1 ICSTOP ,PIDY ,CYCLE ,DT ,NUNSP ,NFRPLP ,NDUMP7 ,
2 KUNINTR ,IPR ,TOPMU ,RTMU ,NOGAS ,NUMREZ ,ETH ,
3 IGM ,SLPNDX ,PRCNT ,KUNITW ,IGROW ,IDLT ,NUN20 ,
4 JDBT ,JDTIP ,SLPNDY ,DMIN ,NINAX ,DTNA ,CVIS ,
5 JMAX ,JDBT ,NC ,IDRT ,NUN32 ,IMAX ,NUN34 ,
6 MAPS ,NUN36 ,KNAX ,ICLADD ,BOTM ,BOTMV ,NUNSP ,
7 IPCYCL ,TSTOP ,PRLIM ,PRDEL ,PRFACT ,I1 ,I2 ,
8 IPLGRT ,RTH ,WFLAGF ,WFLAGL ,PLWMIN ,IPLGBT ,UN65 ,
9 TOPM ,BOTHU ,UN66 ,RTW ,TOPMV ,NUN67 ,NNAI ,GAMA ,
COMMON COMMON ,CYPH3 ,UN71 ,NTRACR ,NMXCLS ,BBOUND ,ALPHA ,
1 ECK ,NECYCL ,NTPMX ,UN79 ,INTER ,NTCC ,STEMIN ,
2 IVARDX ,I ,EMIN ,PMIN ,UN94 ,REZ ,NODUMP ,
3 K ,UN98 ,MOVIE ,EVAPM ,EVAPN ,EVAPMU ,EVAPMV ,
4 UN97 ,UN98 ,NSLPMX ,NSLPMN ,NENDMI ,NVRTEX ,ROEPS ,
5 EZPH2 ,NLINER ,NSLPMX ,NSLPMN ,NENDMI ,NVRTEX ,ROEPS ,
6 PLGOPT ,NSLD ,FINAL ,UN114 ,NOSLIP ,LVISC ,NUN117 ,
7 NADD ,MINX ,MAXC ,MINY ,MAXY ,IEXTX ,JEXTY ,
8 PLUGON ,UN126 ,UN127 ,UN128 ,UMIN ,UN130 ,PRIME ,
9 EOR ,EOT ,EOB ,EMOR ,DXF ,DYF ,SALPHA ,
COMMON COMMON ,STAB ,XIENRG ,XENRG ,XTENRG ,UN143 ,DTMIN ,
1 NUN146 ,EMOT ,NUN147 ,CRATIO ,BBAR ,EMOB ,
2 PK(5)

NON-DIMENSIONED VARIABLES

COMMON CYC ,ENERGY ,ERDUMP ,I3 ,KA ,KR ,
1 MA ,HFK ,MO ,MOS ,MR ,NERR ,
2 NK ,NPRINT ,NR ,NVOID ,PRESUR ,PVRTEX ,
3 RHOW ,RHOM ,SDT ,TWOPI ,TXCL ,TYCL ,
4 WS ,WSA ,WSB ,WSC ,WSX ,WSY ,
5 ZMOM ,LAST ,WSC ,WSX ,WSY ,

COMMON/TRACKS/ XP(1800),YP(1800)

COMMON/MXCELL/ RH0Z(30),CNAUT(30),MAT(30)
COMMON/MXCELL2/SAMMY(4,600),SGAMC(4,600),SAMPY(4,600),
2 SAMP(4,600)
LEVEL 2,SAMMY,SGAMC,SAMPY,SAMP

COMMON/SL/ SAMUT(4),SAMVT(4),SAMUR(4),SAMVR(4),SAMMU(4),SAMMV(4),
2 SSIGMU(4),SSIGMV(4)

COMMON/SLIDE/ MSLD(300),XINT(2,300),YINT(2,300)

COMMON/SLIDPK/ NSLAVD(4),MASTRD(4),NENDMD(4),NBGM(4),NENDSD(4),
2 NBGSD(4)

COMMON/MISC/ PQ(5,8),XAXIS(5,75),YAXIS(5,150),VL(4)

```


107	FIT2		
108	FIT2		
109	FIT2		
110	FIT2		
111	FIT2		
112	FIT2		
113	FIT2		
114	FIT2		
115	FIT2		
116	FIT2		
117	FIT2		
118	FIT2		
119	FIT2		
120	FIT2		
121	FIT2		
122	FIT2		
123	FIT2		
124	FIT2		
125	FIT2		
126	FIT2		
127	FIT2		
128	FIT2		
129	FIT2		
130	FIT2		
131	FIT2		
1	LCMFI		
2	LCMFI		
132	FIT2		
133	FIT2		
134	FIT2		
135	FIT2		
136	FIT2		
137	FIT2		
138	FIT2		
139	FIT2		
140	FIT2		
141	FIT2		
142	FIT2		
143	FIT2		
144	FIT2		
145	FIT2		
146	FIT2		
147	FIT2		
148	FIT2		
149	FIT2		
150	FIT2		
151	FIT2		
152	FIT2		
153	FIT2		
154	FIT2		
155	FIT2		
156	FIT2		
157	FIT2		
158	FIT2		
159	FIT2		

C	COMMON/INFACE/	FRACTP(5,600)
C	COMMON/PH2/	FLEFT(150),YAMC(150),GAMC(150),SIGC(150),SDELET(4),SDELER(4),SDELEB(4),SDELM(4),SSIGC(4,150),UTL(4,150),VTL(4,150),UTRANS(4,4),VTRANS(4,4),SYAMC(4,150),SFLEFT(4,150),SDLKEL(4,150),SDLKER(4),SDLKEB(4),SDLKET(4),DLKEL(150)
C	COMMON/REZONE/	XMAS(2,4),SSIE(2,4),XRH(2,4),USSAVE(2,4),VSSAVE(2,4),THEI(2),MFGREZ(150),REZAMX(150),REZAIX(150),REZOMS(4,150),REZSIE(4,150),REZRHO(4,150),STKEGM(2,4)
C	***TKEG	IS SPECIFIC KINETIC ENERGY IN A CELL
C	***TKEGM	IS SPECIFIC KINETIC ENERGY OF MATERIALS IN INTERFACE CELLS
C	LEVEL 2	TKEG,TKEGM
C	COMMON/KEGCHK/TKEG(8001)	,TKEGM(4,600)
C	COMMON/ERRDEC/	IRAY(6)
C	DIMENSION VOLM(5,600)	
C	DIMENSION XO(2),YO(2),DXO(2),DYO(2)	
C	DIMENSION CYCLEI(40)	
C	LEVEL 2	XP,YP,MSLD,XINT,YINT,PQ,XAXIS,YAXIS,VL,PLUP,PLWC,PVOL,ALFSV,X00,X,Y00,Y,DX00,DX,DY00,DY
C	EQUIVALENCE (Z(1),PROB).	(PIDY,PI)
C	SPECIAL EQUIVALENCE FOR SETUP	
C	EQUIVALENCE (SAMMY,VOLM)	
C	EQUIVALENCE (XO(2),X(1)),(YO(2),Y(1)),(DXO(2),DX(1)),(DYO(2),DY(1))	
C	EQUIVALENCE(CYCLEI(1),CYCLE)	
C	DATA STATEMENTS FOR MATERIALS DEFINED IN EQST.	
C	NOTE - IDEAL GAS REFERENCE DENSITY RHOZ(20) IS DEFINED TO BE THE NORMAL DENSITY OF AIR. THE USER SHOULD CHANGE THIS VALUE IF USING ANOTHER MATERIAL.	
C	DATA (RHOZ(K),K=1,19) /19.17,8.9,7.8,2.79,1.8	
C	1 4.6,8.9,10.2,11.7,11.3	
C	2 9,2.7,2.7,1.97,1.7	
C	3 2.3,2.8,2.7,2.2,2.2	
C	DATA RHOZ(20) / .001293/	
C	DATA (RHOZ(K),K=21,25)/1.717,1.63,1.821,1.84,1.608/	
C	DATA (CNAUT(K),K=1,19) /4.01E5,3.96E5,4.03E5,5.27E5,8.06E5,4.78E5,4.63E5,5.16E5,2.13E5,2.03E5	
C	1	

```

2      2.89E5 ,2.68E5 ,2.58E5 ,2.24E5 ,1.63E5 /
3      3.49E5 ,5.51E5 ,3.85E5 ,3.37E5 /
C*****CALL SYSTEMC TO INHIBIT PRINTING OF ERROR 115 AND
C      START ERROR SUMMARY ACCUMULATION.
C      DATA IRAY/6*( 0)/
C      CALL SYSTEMC (115,IRAY)
C*****END COMMON BLOCK FOR MRLHELP... *****
C      MOVTCR 7
C      MOVTCR 8
C      MOVTCR 9
C      MOVTCR 10
C      MOVTCR 11
C      MOVTCR 12
C      MOVTCR 13
C      MOVTCR 14
C      MOVTCR 15
C      MOVTCR 16
C      MOVTCR 17
C      MOVTCR 18
C      MOVTCR 19
C      MOVTCR 20
C      MOVTCR 21
C      MOVTCR 22
C      MOVTCR 23
C      MOVTCR 24
C      MOVTCR 25
C      MOVTCR 26
C      MOVTCR 27
C      MOVTCR 28
C      MOVTCR 29
C      MOVTCR 30
C      MOVTCR 31
C      TRLOGIC 1
C      MOVTCR 32
C      TRLOGIC 2
C      MOVTCR 33
C      MOVTCR 34
C      MOVTCR 35
C      MOVTCR 36
C      MOVTCR 37
C      MOVTCR 38
C      MOVTCR 39
C      MOVTCR 40
C      MOVTCR 41
C      MOVTCR 42
C      MOVTCR 43
C      MOVTCR 44
C      MOVTCR 45
C      MOVTCR 46
C      MOVTCR 47
C      MOVTCR 48
C      MOVTCR 49
C      MOVTCR 50
C      MOVTCR 51
C*****
C      NVP=NVOID
C      NV=NMP(NVOID)-1
C      *** WHEN NTCC.GT.1, PASSIVE CELL CENTERED TRACERS (XP,YP)
C      ARE BEING PROCESSED.
C      WHEN NTCC=0, THESE TRACERS HAVE NOT BEEN GENERATED.
C      D0 1530 N=1,NVP
C      NN=NMP(N)-1
C      IF (N.EQ.NVOID) NN=NTCC
C      IF (NN.EQ.0) GO TO 1530
C      *** CELL-CENTERED TRACERS MOVED ONLY ONCE PER CYCLE
C      IF (N.GT.NMAT.AND.LJ.LT.ICY) GO TO 1630
C      D0 1620 LL=1,NN
C      L=NN-LL+1
C      *** FIND I AND J OF CELL TRACER IS IN BEFORE IT IS MOVED.
C      IF (N.LT.NVP) GO TO 1180
C      I=INT(XP(L))+1
C      IF(XP(L).GE.FLOAT(IMAX)) I=I-1
C      J=INT(YP(L))+1
C      IF(YP(L).GE.FLOAT(JMAX)) J=J-1
C      GO TO 1230
C      CONTINUE
C      IF (TX(N,L).LE.-1000.) GO TO 1620
C      I=INT(TX(N,L))+1
C      IF (TX(N,L).GE.FLOAT(IMAX)) I=I-1
C      J=INT(TY(N,L))+1
C      IF (TY(N,L).GE.FLOAT(JMAX)) J=J-1
C      1180
C      *** FIND INDEX OF CORRESPONDING FREE SURFACE TRACER
C      IF THERE IS ONE.
C      LVF=0
C      IF (NV.LE.0) GO TO 1220
C      D0 1200 LLV=1,NV
C      LV=NV-LLV+1
C      IF (TX(N,L)-TX(NVOID,LLV)) 1200,1190,1200
C      IF (TY(N,L)-TY(NVOID,LLV)) 1200,1210,1200
C      1190

```

1200	CONTINUE	MOVTCR	52
	GO TO 1220	MOVTCR	53
1210	LVF=LV	MOVTCR	54
C		MOVTCR	55
1220	CONTINUE	MOVTCR	56
C	*** IF TRACER IS OUTSIDE GRID, SKIP OUT.	MOVTCR	57
1230	CONTINUE	MOVTCR	58
	IF (I.GT. I1 OR J.GT. I2 OR J.LT. 0) GO TO 1520	MOVTCR	59
C		MOVTCR	60
	K=(J-1)*IMAX+I+1	MOVTCR	61
	M=IABS(NFLAG(K))-100	MOVTCR	62
C		MOVTCR	63
C	*** STORE FRACTIONAL PARTS OF THE COORDINATES IN FRX, FRY.	MOVTCR	64
		MOVTCR	65
C		MOVTCR	66
	IF (N.LT. NVP) GO TO 1240	MOVTCR	67
	FRX=XP(L)-AINT(XP(L))	MOVTCR	68
	FRY=YP(L)-AINT(YP(L))	MOVTCR	69
C	THIS LOGIC CHANGED BY DLS 30.8.83 TO ACCOUNT FOR TRACERS	TRLOGIC	3
C	ON FAR EDGE OF LAST ROW OR COLUMN, WHICH WERE PREVIOUSLY	TRLOGIC	4
C	TREATED AS BEING ON NEAR EDGE, AS FRX WAS 0.0	TRLOGIC	5
	IF (XP(L).GE.FLOAT(IMAX)) FRX=1.0	TRLOGIC	6
	IF (YP(L).GE.FLOAT(JMAX)) FRY=1.0	TRLOGIC	7
	GO TO 1250	MOVTCR	70
1240	CONTINUE	MOVTCR	71
	FRX=TX(N,L)-AINT(TX(N,L))	MOVTCR	72
	FRY=TY(N,L)-AINT(TY(N,L))	MOVTCR	73
	IF (TX(N,L).GE.FLOAT(IMAX)) FRX=1.0	TRLOGIC	8
	IF (TY(N,L).GE.FLOAT(JMAX)) FRY=1.0	TRLOGIC	9
1250	CONTINUE	MOVTCR	74
	WSX=ABS(FRX-.5)	MOVTCR	75
	WSY=ABS(FRY-.5)	MOVTCR	76
C		MOVTCR	77
C	*** DEFINE INDICES OF NEIGHBOR CELLS.	MOVTCR	78
	KV=K+IMAX	MOVTCR	79
	IF (FRY.GE..5 AND J.EQ. JMAX) GO TO 1260	MOVTCR	80
	IF (FRY.GE..5) GO TO 1280	MOVTCR	81
	IF (J.GT. 1) GO TO 1270	TRLOGIC2	1
C	NOTE NEXT COMMENT NOT TRUE (AND NEVER WAS!) HENCE THIS FIX	TRLOGIC	11
C	*** FRY=0. MEANS POINT IS ON TOP OR BOTTOM GRID BOUNDARY	MOVTCR	83
1260	KV=K	MOVTCR	84
	GO TO 1280	MOVTCR	85
1270	KV=K-IMAX	MOVTCR	86
C		MOVTCR	87
1280	KH=K+1	MOVTCR	88
	KD=KV+1	MOVTCR	89
	IF (FRX.GE..5 AND I.EQ. IMAX) GO TO 1290	MOVTCR	90
	IF (FRX.GE..5) GO TO 1310	MOVTCR	91
	IF (I.GT. 1) GO TO 1300	TRLOGIC2	2
C	NOTE NEXT COMMENT NOT TRUE (AND NEVER WAS!) HENCE THIS FIX	TRLOGIC	13
C	*** FRX=0. MEANS POINT IS ON LEFT OR RIGHT GRID BOUNDARY	MOVTCR	94
1290	KH=K	MOVTCR	95
C	BRL VERSION HAD ERROR HERE OF KD=K	TRLOGIC	14
	KD=KV	TRLOGIC	15

```

GO TO 1310
KH=K-1
KD=KV-1
C
1310 CONTINUE
C
JV=(KV-2)/IMAX+1
JH=(KH-2)/IMAX+1
IH=KH-(JH-1)*IMAX-1
C
C
*** DEFINE WEIGHTING FACTORS
WH=0.
WV=0.
WD=0.
WK=0.
WFS=0.
WH=WSX*(1.0-WSY)*AMX(KH)/(TAU(IH)*DY(J))
WV=WSY*(1.0-WSX)*AMX(KV)/(TAU(I)*DY(JV))
WD=WSX*WSY*AMX(KD)/(TAU(IH)*DY(JV))
WK=(1.0-WSY)*(1.0-WSX)*AMX(K)/(TAU(I)*DY(J))
C
WF(1)=WK
WF(2)=WV
WF(3)=WD
WF(4)=WH
KI(1)=K
KI(2)=KV
KI(3)=KD
KI(4)=KH
MTK(1)=M
MTK(2)=IABS(MFLAG(KV))-100
MTK(3)=IABS(MFLAG(KD))-100
MTK(4)=IABS(MFLAG(KH))-100
C
C
*** IF THIS IS AN IMPACT CALCULATION WITH PLUGGING
FAILURE, COMPUTE VELOCITY COMPONENTS OF PROJECTILE,
PLUG AND TARGET TRACERS IN SUBROUTINE PLUGUV
C
IF(PLUGON.LE.0. .OR. N.GT.3) GO TO 1305
LS=L
CALL PLUGUV
GO TO 1360
C
1305 CONTINUE
C

```

```

C      WFS=WF(1)+WF(2)+WF(3)+WF(4)
C      *** IF WFS=0. THERE IS NO MASS IN THE OVERLAP CELLS. SO
C      THAT THIS PARTICLE WILL NOT BE LEFT BEHIND IT IS MOVED
C      TO THE POSITION OF THE L+1 PARTICLE UNLESS L=NN, IN
C      WHICH CASE IT IS MOVED TO THE POSITION OF THE L-1
C      PARTICLE. IN ANY CASE, IF IT IS ON A GRID LINE IT IS
C      ONLY MOVED ALONG THAT GRIDLINE.
C
C      IF(N.EQ.NVP .OR. WFS.GT.O.) GO TO 1335
C      NT=1
C      THIS FIX BY DLS TO PREVENT TRACERS 1 AND NN GETTING STUCK.
C      IT APPEARS THIS WAS INTENDED ORIGINALLY, AT LEAST FOR THE
C      L=NN CASE. THE PROBLEM BECAME MORE LIKELY TO ARISE WITH
C      THE USE OF THE DEGAS ROUTINE TO ISOLATE PACKAGE ONE.
C
C      IF(L.EQ.NN)NT=-1
C      IF(TX(N,L+NT).LE.-1000.)GO TO 1505
C      ***NOTE PREVIOUSLY THE PROG SKIPPED TO 1505 IF L WAS 1 OR NN.
C      ***ALSO TX FOR L+1 AND L-1 WERE TESTED, WITHOUT USING NT=-1.
C
C      IF(TX(N,L).GT.O. .AND. TX(N,L).LT.FLOAT(IMAX)) TX(N,L)=TX(N,L+NT)
C      IF(TY(N,L).GT.O. .AND. TY(N,L).LT.FLOAT(JMAX)) TY(N,L)=TY(N,L+NT)
C      GO TO 1505
C
C      1335 CONTINUE *** CALCULATE RADIAL VELOCITY OF THE TRACER.
C
C      IF (I.GT.1.OR.FR.XE.(.5)) GO TO 1340
C      WF(3)=-WF(3)
C      WF(4)=-WF(4)
C
C      1340 CONTINUE
C      UEFF=(UTK(1)*WF(1)+UTK(2)*WF(2)+UTK(3)*WF(3)+UTK(4)*WF(4))/WFS
C      WF(3)=ABS(WF(3))
C      WF(4)=ABS(WF(4))
C
C      *** CALCULATE AXIAL VELOCITY OF THE TRACER.
C      IF (J.GT.1.OR.CVIS.LT.O..OR.FRY.GE.(.5)) GO TO 1350
C      WF(2)=-WF(2)
C      WF(3)=-WF(3)
C
C      1350 CONTINUE
C      VEFF=(VTK(1)*WF(1)+VTK(2)*WF(2)+VTK(3)*WF(3)+VTK(4)*WF(4))/WFS
C
C      1360 CONTINUE
C      IF (ABS(UEFF).LT.UMIN) UEFF=0.
C      IF (ABS(VEFF).LT.UMIN) VEFF=0.
C
C      *** CELL-CENTERED TRACERS MOVED ONLY ONCE PER CYCLE
C      IF (N.GT.NMAT.AND.L.J.EQ.ICY) SDT=DT
C      DISTX=UEFF*SDT
C      DISTY=VEFF*SDT
C      POSX=X(I-1)+DX(I)*FRX+DISTX
C      POSY=Y(J-1)+DY(J)*FRY+DISTY
C
C      IF (N.LT.NVP) GO TO 1500
C      *** DO NOT MOVE TRACERS OFF OF GRID BOUNDARIES
C      IF(XP(L).GT.O. .AND. XP(L).LT.FLOAT(IMAX)) XP(L)=XPLOC(POSX)
C      IF(YP(L).GT.O. .AND. YP(L).LT.FLOAT(JMAX)) YP(L)=YPLOC(POSY)

```

```

199      MOVTCR
200      MOVTCR
201      MOVTCR
202      MOVTCR
203      MOVTCR
204      MOVTCR
205      MOVTCR
206      MOVTCR
207      MOVTCR
208      MOVTCR
209      MOVTCR
210      MOVTCR
211      MOVTCR
212      MOVTCR
213      MOVTCR
214      MOVTCR
215      MOVTCR
216      MOVTCR
217      MOVTCR
218      MOVTCR
219      MOVTCR
220      MOVTCR
221      MOVTCR
222      MOVTCR
223      MOVTCR
224      MOVTCR
225      MOVTCR
226      MOVTCR
227      MOVTCR
228      MOVTCR
229      MOVTCR
230      MOVTCR
231      MOVTCR
232      MOVTCR
233      MOVTCR
234      MOVTCR
235      MOVTCR
236      MOVTCR
237      MOVTCR
238      MOVTCR
239      MOVTCR
240      MOVTCR
241      MOVTCR
242      MOVTCR
243      MOVTCR
244      MOVTCR
245      MOVTCR
246      MOVTCR
247      MOVTCR
248      MOVTCR
249      MOVTCR
250      MOVTCR
251      MOVTCR
252      MOVTCR

      IF (XP(L).LT.O.) XP(L)=O.
      IF (YP(L).LT.O.) YP(L)=O.

      GO TO 1510
      *** DO NOT MOVE TRACERS OFF OF GRID BOUNDARIES ***
      1500 IF (TX(N,L).GT.O. .AND. TX(N,L).LT.FLOAT(IMAX)) TX(N,L)=XPTOC(POSX)
      IF (TY(N,L).GT.O. .AND. TY(N,L).LT.FLOAT(JMAX)) TY(N,L)=YPTOC(POSY)
      *** IF TRACER CROSSED AXIS, PUT IT BACK ON AXIS.
      IF (TX(N,L).LT.O.) TX(N,L)=O.
      IF (TY(N,L).LT.O.) TY(N,L)=O.
      IF (TX(N,L).GT.FLOAT(IMAX)) TX(N,L)=FLOAT(IMAX)
      IF (TY(N,L).GT.FLOAT(JMAX)) TY(N,L)=FLOAT(JMAX)
      IF (N.EQ.NLINER.AND.TX(N,L).LE.O..AND.L.GT.1.AND.L.LT.NN) TX(N,L)=
1.0001
      *** MOVE CORRESPONDING FREE SURFACE TRACER IF THERE IS ONE
      1505 IF (LVF.EQ.O) GO TO 1510
      TX(NVOID,LVF)=TX(N,L)
      TY(NVOID,LVF)=TY(N,L)

      1510 CONTINUE
      *** END OF LOOP ON L
      1520 CONTINUE
      *** END OF LOOP ON MATERIALS (MN)
      1530 CONTINUE

      IF (PLGOPT.LE.O.) GO TO 1535
      *** BEFORE PLUG BEGINS TO FORM, SLPNDX AND SLPNDY
      ARE DEFINED TO BE EQUAL TO THE COORDINATES OF
      THE TRACER AT THE BEGINNING OF THE SLOPLINE.
      IF (PLUGON.GT.O.) GO TO 1535
      NB=NBGSD(2)
      SLPNDX = TX(2,NB)
      SLPNDY = TY(2,NB)
      1535 CONTINUE
      *** CLOSE VOID IF NVRTEX.GT.O. NVRTEX IS THE NUMBER OF THE
      VOID TRACER WHICH IS THE VERTEX OF THE REGION TO BE
      CLOSED
      IF (NVRTEX.GT.O) CALL VDCLOS

      *** AFTER THE PLUG IS COMPLETELY FORMED, ALL THE SLIDE
      TRACERS ARE MOVED WITH PLUG AND PROJECTILE VELOCITIES
      EXCEPT THE TOP TRACER. THE TRACERS BEHIND THE TOP
      POINT WILL USUALLY HAVE A LARGER AXIAL COMPONENT OF
      VELOCITY AND WILL EVENTUALLY TRY TO PASS THE TOP
      TRACER. SUBROUTINE ENDMV IS CALLED TO PREVENT THE
      TRACERS BEHIND FROM BYPASSING THE TOP TRACER.

      IF (PLGOPT.LE.O.) RETURN
      IF (TX(3,1).GT.O.) CALL ENDMV
      RETURN
      END

```


APPENDIX 4

AN UPDATE RUN TO ADD SUBROUTINE TCRCHK
TO THE HELP CODE

```

MAKE21,P1000,T100.
GETSET,DFC4405.
ATTACH,OLDPL,MRLHLP20,SN=DFC4405,ID=DFCDLS.
REQUEST,NEWPL,*PF,SN=DFC4405.
REQUEST,LGO,*PF,SN=DFC4405.
UPDATE,F,N,L=A1.
FTN,D=TAPE33,I=COMPILE,SL=0,LCM=1.
CATALOG,NEWPL,$MRLHLP21$,SN=DFC4405,ID=DFCDLS,RP=998.
CATALOG(LGO,$LGO FOR MRLHLP21$,SN=DFC4405,ID=DFCDLS,RP=998)
*EOS
*IDENT SPEEDUP
*DELETE FIT2.71
      3      DETIM,          COINC
*DELETE FIT2.85
      4      OUTIE(4),OUTKE(4),DETIM(8001),COINC(1800)
*INSERT ADDTCR.27
      NADDED=0
*INSERT ADDTCR.93
      NADDED=NADDED+NADD
*INSERT ADDTCR.179
      WRITE(6,301)NADDED
      IF(NADDED.LE.0)GOTO 300
      CALL TCRCHK
300      CONTINUE
301      FORMAT(//,1X,I5,2X,13HTRACERS ADDED)
*INSERT MAIN.17
C      FILL COINC ARRAY TO CHECK FOR COINCIDENT TRACERS ON SETUP
C      OR RESTART OF PROBLEM
      CALL TCRCHK
*DELETE MOVTCR.41,56
*DELETE MOVTCR.213,217
1505      CONTINUE
*INSERT MOVTCR.222
C      MOVE FREE SURFACE TRACERS
C      NOTE ALL VOID TRACERS ARE NOW FLAGGED TO A TRACER OF ANOTHER
C      MATERIAL. ARRAY COINC HOLDS THIS INFO, PACKAGE NUMBER AND MATERIAL
C      NUMBER PACKED INTO ONE WORD. SUBROUTINE TCRCHK UPDATES THIS ARRAY
C      AS REQUIRED.
      IF(NV.LE.0)GOTO 2001
      DO 2000 I=1,NV
      CC=COINC(I)
      IF(CC.LE.0.0)GOTO 2000
      NPACK= INT((CC-AINT(CC))*10.0)
      NTCR= INT(CC)
      TX(NVOID,I)=TX(NPACK,NTCR)
      TY(NVOID,I)=TY(NPACK,NTCR)
2000      CONTINUE
2001      CONTINUE
*INSERT VDCLOS.140
      CALL TCRCHK
*ADDFILE
*DECK TCRCHK
      SUBROUTINE TCRCHK
*CALL COMDK
C      THIS SUBROUTINE CHECKS FOR COINCIDENT TRACERS WITH THE VOID
C      PACKAGE. THIS INFORMATION IS NEEDED BY SUBROUTINE MOVTCR,

```

C
C
C
C
C
C
C
C
C
C

WHICH PREVIOUSLY RE-CHECKED EVERY CYCLE, EVEN WHEN NO CHANGES WERE POSSIBLE FROM THE PREVIOUS CHECKS.

TCRCHK IS CALLED BY MAIN AFTER SETUP IS CALLED, EITHER FOR A NEW PROBLEM OR A RESTART.

TCRCHK IS ALSO CALLED BY ADDTCR WHEN NEW TRACERS ARE ADDED AND BY VDCLOS WHEN A VOID IS CLOSED.

WRITTEN BY DAVID L. SMITH APRIL 1984.

NV=NMP(NVOID)-1
IF(NV.LE.0)RETURN
DO 100 NN=1,NV
COINC(NN)=0.0
TXV=TX(NVOID,NN)
IF(TXV.LE.-1000.)GOTO 100
TYV=TY(NVOID,NN)
DO 110 NPACK=1,NMAT
NT=NMP(NPACK)-1
DO 120 NTCR=1,NT
IF(TXV.EQ.TX(NPACK,NTCR).AND.TYV.EQ.TY(NPACK,NTCR))GOTO 130
120 CONTINUE
110 CONTINUE
GOTO 100
130 COINC(NN)=FLOAT(NTCR)+(FLOAT(NPACK)+.001)/10.0
100 CONTINUE
WRITE(6,200)
RETURN
200 FORMAT(//,34H SUBROUTINE TCRCHK HAS BEEN CALLED,//)
END
*EOP

(MRL-R-925)

DISTRIBUTION LIST

MATERIALS RESEARCH LABORATORIES

Director
Superintendent, Physical Chemistry Division
Mr F.G.J. May
Dr B.W. Thorpe
Library (2 copies)
David L. Smith

DEPARTMENT OF DEFENCE

Chief Defence Scientist (for CDS/DCDS/CPAS/CERPAS) (1 copy)
Army Scientific Adviser
Air Force Scientific Adviser
Navy Scientific Adviser
Officer-in-Charge, Document Exchange Centre (18 copies)
Technical Reports Centre, Defence Central Library
Director of Quality Assurance Support (DQAS)
Assistant Director, Defence Scientific and Technical Intelligence,
Joint Intelligence Organisation
Librarian, Bridges Library
Librarian, Engineering Development Establishment
Defence Science Adviser, Australia High (Summary Sheets Only)
Commission, London
Counsellor Defence Science, Washington, D.C. (Summary Sheets Only)
Librarian, (Through Officer-in-Charge), Materials Testing
Laboratories, Alexandria, NSW
Senior Librarian, Aeronautical Research Laboratories
Senior Librarian, Defence Research Centre Salisbury, SA

DEPARTMENT OF DEFENCE SUPPORT

Deputy Secretary, DDS
Head of Staff, British Defence Research & Supply Staff (Aust.)

OTHER FEDERAL AND STATE DEPARTMENTS AND INSTRUMENTALITIES

NASA Canberra Office, Woden, ACT
The Chief Librarian, Central Library, CSIRO
Library, Australian Atomic Energy Commission Research Establishment

MISCELLANEOUS - AUSTRALIA

Librarian, State Library of NSW, Sydney NSW
University of Tasmania, Morris Miller Lib., Hobart, Tas.

(MRL-R-925)

DISTRIBUTION LIST
(continued)

MISCELLANEOUS

Library - Exchange Desk, National Bureau of Standards, USA
UK/USA/CAN/NZ ABCA Armies Standardisation Representative (4 copies)
Director, Defence Research Centre, Kuala Lumpur, Malaysia
Exchange Section, British Library, UK
Periodicals Recording Section, Science Reference Library,
British Library, UK
Library, Chemical Abstracts Service
INSPEC: Acquisition Section, Institute of Electrical Engineers, UK
Engineering Societies Library, USA
Aeromedical Library, Brooks Air Force Base, Texas, USA
Documents Librarian, The Centre for Research Libraries, Chicago, Ill.
Defense Attache, Australian Embassy, Bangkok, Thailand

ADDITIONAL DISTRIBUTION

Mr Don F. Menne, USA Ballistic Research Laboratories (10 copies)
Mr Maurice Laviolette, Defence Research Establishment
Valcartier, Quebec, Canada (2 copies)
Mr Rod T. Schmitke, Defence Research Establishment
Suffield, Alberta, Canada (2 copies)
Dr Peter R. Lee, Royal Armament Research & Development
Establishment, Kent, UK (3 copies)
Dr P.C. Chou, Drexel University, Pennsylvania, USA (3 copies)
Dr M.L. Wilkins, Lawrence Livermore National Laboratory,
California, USA
Dr Klaus Thoma, Bolkow Appautebau GMBH, West Germany (2 copies)

LEAD

FILMED

8

24





